

Programmatic Risk Reduction for Large Software Procurements

**Gary Preckshot
Ed Jones
Dennis Lawrence**

June 1996



This is an informal report intended primarily for internal or limited external distribution. The opinions and conclusions stated are those of the author and may or may not be those of the Laboratory.

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This report has been reproduced
directly from the best available copy.

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (615) 576-8401, FTS 626-8401

Available to the public from the
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.,
Springfield, VA 22161

Programmatic Risk Reduction for Large Software Procurements

by
Gary Preckshot,
Ed Jones,
Dennis Lawrence,
Lawrence Livermore National Laboratory

When making an important decision, program managers must evaluate the risk of each alternative and weigh the trade-offs between one type of risk and another. Program managers are familiar with the concepts of risk analysis and assessment in such areas as human factors, security, and accident prevention. The risk-based approach provides traceable, manageable, and justifiable methods for decision-making.

When acquiring large or complex software systems, program managers can use this same risk-based approach to procure a system that works correctly, is delivered on time and within budget, and meets the needs of the organization. This approach enables managers to assess specific problems or issues, rank their relative importance, determine which have the greatest potential for management, allocate resources appropriately, and monitor the effectiveness of actions taken.

Perspectives of the Procurement Team. Each participant in a software acquisition is likely to calculate risk differently. For instance, the program manager who may be purchasing software or software development services is concerned with meeting the organization's goals and objectives, as determined by upper management. Meanwhile, the software developer may view risk in terms of specific contractual conditions, late access to a market window, or an unprofitable level of maintenance effort [1]. Regulators, on the other hand, focus almost exclusively on protecting workers and the public from unsafe products [2]. Therefore, it is imperative to understand where one fits in this spectrum of risk perception, and to realize that one's viewpoint may not be shared by others involved in the software acquisition.

The software acquisition process should harmonize the differing objectives or, at least, keep them in rational balance.

Factors that Influence Programmatic Risk. To accomplish the programmatic risk reduction described above, the program manager needs to understand the factors that influence both software risk and program risk. Why? The same factors that apply to the software development function can also apply to the software procurement management function. This is because the procurement operation provides the requirements that drive the software developer; an unstable procurement operation results in unstable demands upon the software developer, which translates to increased programmatic risk.

During 1992 and 1993, Lawrence Livermore National Laboratory (LLNL) carried out an investigation on behalf of the U.S. Nuclear Regulatory Commission designed to isolate criteria which could be used by regulators to assess the capability of organizations producing software for safety-critical applications. Data were obtained from a variety of sources, including world-renowned experts, leading companies, software engineering standards and the technical literature. The result is an organized list of “design factors” whose presence provides useful information on the capabilities of software developers. The factors were divided into four graded categories. The first (mandatory positive factors) and last (negative factors) are summarized here; see [3] for additional information.

Seven design factors are considered mandatory; the lack of any one of these factors may be considered sufficient grounds for rejection of a software product. These factors are:

- High-quality management and technical personnel.
- The use of configuration management.
- The existence of clear, stable and validated software requirements.
- The use of independent (of the developers) verification and validation (including testing).
- The use of a formal life cycle for product development.

- Traceability from system requirements and design, through software requirements, software design, code and validation testing.
- The use of hazards analysis and risk analysis to guide software development.

Nine factors were identified whose presence should be cause for caution, or more thorough scrutiny. These negative factors are indicators of an organization in trouble.

- High staff turn-over.
- A history of projects being driven by schedule rather than quality.
- Lack of a sufficiently long organization process history.
- Management which cannot (or does not) enforce stable requirements.
- A history of management's estimates of product reliability greatly exceeding what can be measured or proved.
- A history of failing to meet predicted cost, schedule, and quality goals for products.
- A history of failing to track errors and determine root causes of defects.
- An underfunded development effort.
- A corporate culture that discourages problem-reporting by employees ("kill the messenger" syndrome).

Fortunately, management of the procurement team is within the program manager's control, and may represent the single, most effective risk management tool available for reducing programmatic risk. The design factor study done by LLNL, with slight modifications to adjust the findings to a software procurement operation, should be a valuable management tool for program managers.

An extensive compendium of LLNL's work on software reliability and software risk factors is available in Adobe Acrobat format at <http://nssc.llnl.gov/FESSP/CSRC/CSR.html>. LLNL references listed below can also be found there. Please contact Gary Lynn

Johnson, johnson27@llnl.gov, or the authors
(preckshot1@llnl.gov, jones37@llnl.gov, or lawrence2@llnl.gov)
for more information.

Acknowledgment

This work was performed under the auspices of the U.S.
Department of Energy by Lawrence Livermore National
Laboratory under contract no. W-7405-Eng-48.

References

- [1] Boehm, Barry W. *Software Risk Management*.
- [2] “Software Reliability and Safety in Nuclear Reactor
Protection Systems,” LLNL.
- [3] “Design Factors for Safety-Critical Software,” LLNL.